

How to Setup Application Server to Access DB2 for z/OS with High Availability

M. Sueli Almeida, sueli@us.ibm.com

IBM DB2 for z/OS

Silicon Valley Lab

IBM Information
ON Demand 2009

DB2 z/OS 2009 Technical Conference

1

© 2009 IBM Corporation

This session will describe the technologies that need to be deployed to ensure the highest availability for application servers accessing a DB2 data sharing group. Special focus will be given to the network requirements since the network configuration plays a critical part in the overall DB2 availability story.

Only implementing bits and pieces exposes applications to outages due to a member failure or a member slow down. If configured correctly, application servers will fully utilize the availability and scalability aspects of parallel sysplex.

Once implemented, WLM fully manages resource utilization across the group eliminating the need to reserve capacity on individual members for a specific application or force applications to execute against a single subsystem. At the end of the presentation, an update is provided on some of the exciting new availability features being delivered in the latest DB2 drivers.

Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

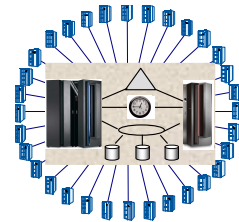
Agenda

- **DB2 z/OS concepts for availability**
- **Client configurations**
- **DB2 Sysplex Workload Balancing**
- **Application server considerations**

DB2 Data Sharing - Availability

Goal: Continuous availability across any planned or unplanned outage across entire z HW/SW stack.

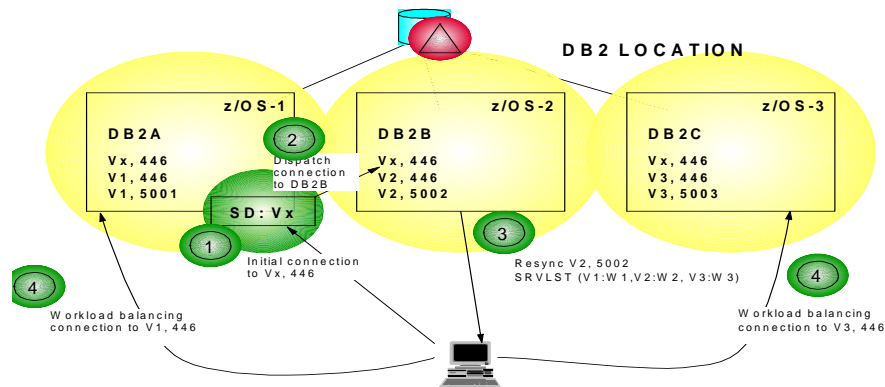
- **Elimination of single point of failure (DB2 subsystem, z/OS system, CPC, I/O paths,..)**
- **Remove all causes for planned outages**
 - rolling “on-line” upgrades and maintenance
 - online schema evolution
 - online utilities
- **On a failure:**
 - Isolate failure to lowest granularity possible
 - Automate recovery and recover fast



Challenge: the distributed application server needs to find the (best) available path to the data.

DB2 z/OS Data Sharing is well acknowledged for providing highest availability, addressing planned and unplanned outages. Nevertheless, an end user only cares that the task is executed, which is not the case if the application server can not reach out to the database server even the database server is up and running. The presentation walks through the step-by-step process to setup a distributed application environment accessing a DB2 Data Sharing group for high availability. It discusses frequently asked questions around sysplex workload balancing and connection concentration.

DVIPA and Sysplex Distributor – the Concept



- **Group DVIPA provides a virtual TCP/IP address into the Data Sharing Group**
- **Sysplex Distributor routes the connection request to the most available member based on WLM recommendation**

The first critical component is the TCP/IP Sysplex Distributor. It provides a fault tolerant network environment by allowing connections always to work to the group as long as one member is listening on the common port. The Sysplex Distributor provides connection level load distribution and provides a single image to the remote applications.

Balancing Connections across group:

- The TCP/IP Sysplex Distributor is used to establish the initial connection to the DB2 group
 - IP address used by clients to access DB2 is configured as a distributed dynamic VIPA
- Provides connection load distribution
- Ensures highest availability possible
- But...
 - Typical connections used for SQL have a long lifespan. For example, an application server that utilizes connection pooling...

DVIPA and Sysplex Distributor – Consideration

- Benefit:
 - **Connections are successful as long as one member is up**
 - Connection level workload balancing between members
 - Setup is isolated to z/OS environment

- Drawbacks
 - SD on one lpar may route to a member on a different lpar, which results into slightly higher response time compared to direct member access.
 - Information about availability of data sharing members is only considered at creation of a "new" connection but application server typically maintain long-running connections.

- ✓ **DVIPA and Sysplex Workload Balancing at the application server need to be enabled to ensure the highest availability characteristics in a distr. environment.**

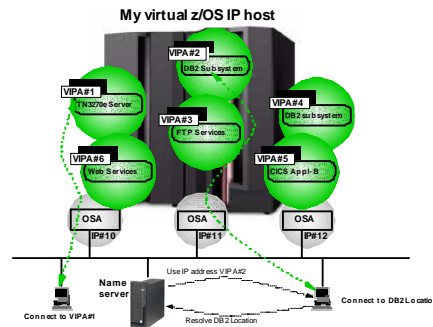
We get frequently the question if there is any additional benefits in using sysplex workload balancing at the application server / DB2 Connect Gateway if we use a group DVIPA and sysplex distributor on z/OS?

Yes, both DVIPA/sysplex distributor on z/OS and Sysplex Workload Balancing on distr. need to be enabled to ensure highest availability characteristics.

High Availability Configuration for Remote Access

- **TCP/IP Sysplex Distributor**
 - Needs to be configured at the network layer and used for the initial connections to the DB2 group
- **DB2 Sysplex Workload Balancing**
 - Needs to be enabled on the application serving platform
- **Load distribution managed by WLM**

The network view of a Parallel Sysplex
 • a single large server with many network interfaces and many services like a DB2 data sharing group



DB2 high availability configuration has three major components:

The first critical component is the **TCP/IP Sysplex Distributor**. It provides a fault tolerant network environment by allowing connections always to work to the group as long one member is listening on the common port. The Sysplex Distributor provides connection level load distribution and provides a single image to the remote applications.

The next critical component is the **DB2 Sysplex Workload Balancing** provided by the DB2 Connect Server drivers. The DB2 Sysplex Workload Balancing function provides transaction level load distribution.

Both components must be enabled and configured correctly to ensure the highest availability characteristics. The next few charts will go into more detail on these technologies and describes why they are critical.

Balancing transactions across group

- **DB2 Sysplex WLB provides transaction level load distribution**
 - Each DB2 member is configured with a unique dynamic VIPA and automatic VIPA takeover
- **DB2 Sysplex WLB is enabled on the application server platform**
- **WLM ensures work is routed to the member with the most capacity and best health**
- **Superior utilization of group resources**

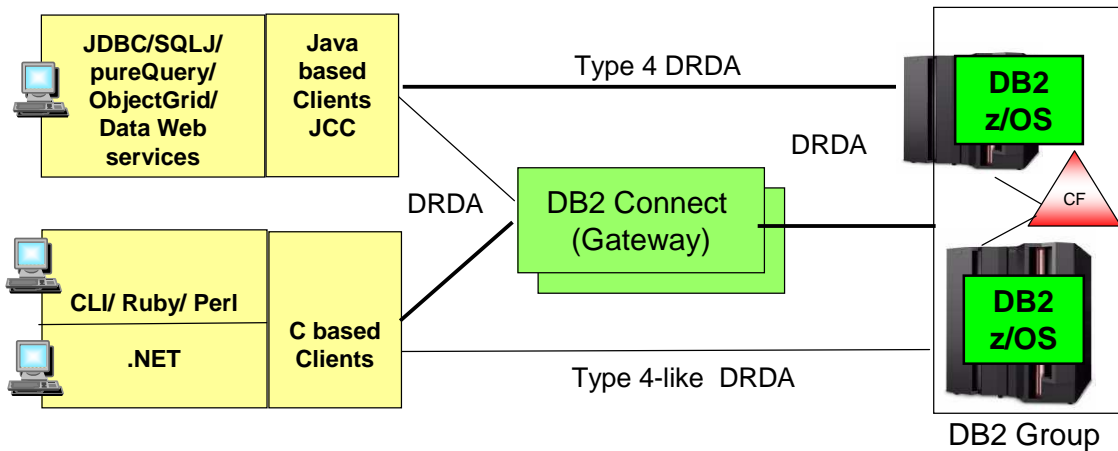
When using connection pooling, the connection is only available for reuse after the application owning the connection issues a disconnect request. In many client-server applications, users do not disconnect for the duration of the workday. Likewise, most application servers establish database connections at start up and do not release these connections until the application server is shut down. In these environments, connection pooling has little, if any, benefit. However, in web and client-server environments where the frequency of connections and disconnections is higher then connection pooling will produce significant performance benefits.

DB2 Sysplex Workload Balancing function supported by DB2 Connect Server eliminates this problem. The DB2 Sysplex Workload Balancing uses connections to DB2 group only for the duration of an SQL transaction while keeping user application's connections logically active. The next chart will go into detail on how this works. This allows for configurations where the number of DB2 threads and the resources they consume can be much smaller than if every application connection had its own active DB2 thread.

This function must be enabled on the DB2 Connect Server. It uses the same WLM recommendations as the sysplex distributor to distribute transactions across the group.

When it comes to fail-safe operation and load balancing of work, sysplex workload balancing is clearly the only choice as it allows reallocation of work with every new transaction.

DB2 Client Configuration to Access DB2 z/OS



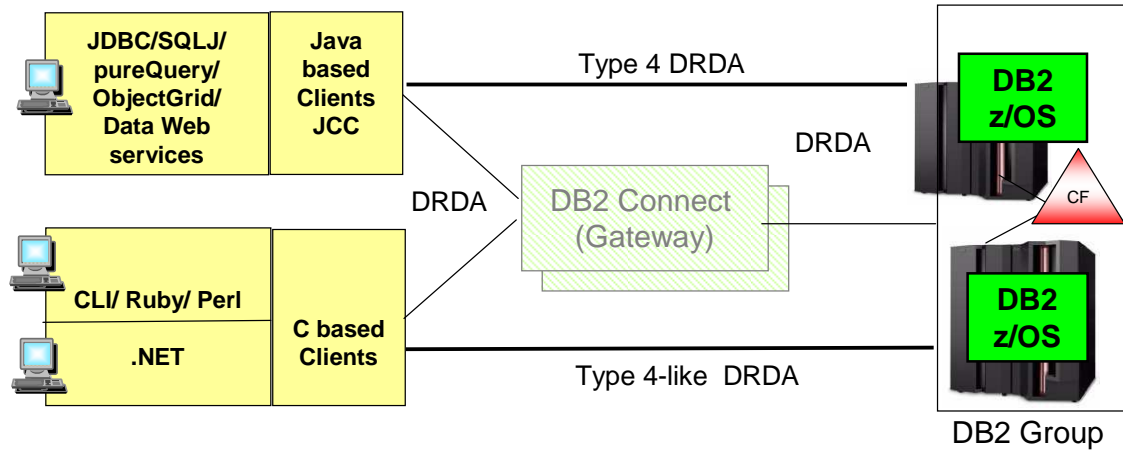
- Until DB2 Connect V7, DB2 Connect converted DB2RA (distr. protocol for DB2 LUW) into DRDA (distr. protocol for DB2 z/OS, DB2 iSeries)
- All C based application connect via DB2 Connect (client or gateway) or type 4 – like with IBM Data Server Driver 9.5
- Java based application connect directly via JCC type 4 or via DB2 Connect Gateway.

IBM Data Server Driver 9.5 for CLI and .NET App

- **Smaller footprint and stand alone download for driver and development component (11M+30M)**
- **run DB2 Connect EE or PE license manager to install license entitlement for connectivity to DB2 z/OS**
- **Sysplex Workload Balancing support in Data Server Driver 9.5 FP3**
- **More detailed instructions**

<http://www.ibm.com/developerworks/wikis/display/DB2/DB2%20and%20.NET%20FAQ>

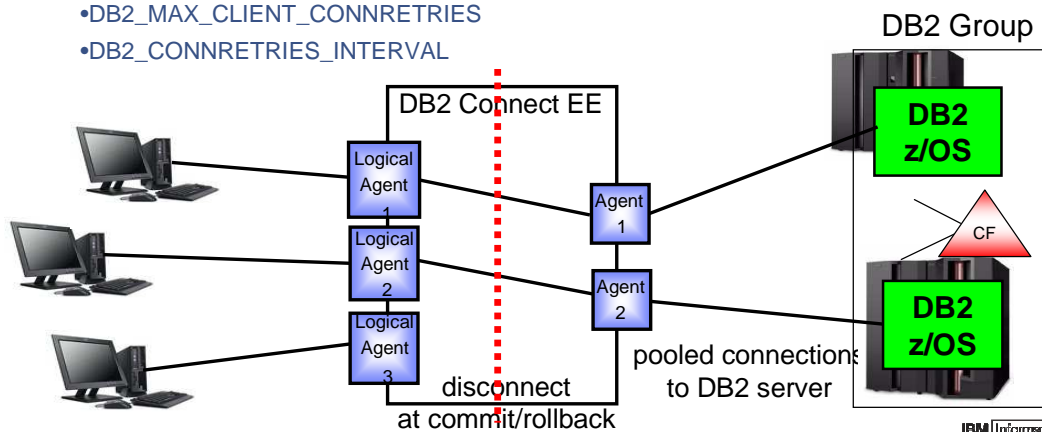
DB2 Client Configuration to Access DB2 z/OS – Future Directions



- Clients - especially application server - access DB2 z/OS directly (will not change licensing model)
- Sysplex Workload Balancing is supported by DB2 Connect gateway, JCC type 4, and Data Server Driver 9.5 FP3 for CLI and .NET application

DB2 Connect EE Connection Concentrator

- **MAXAGENTS** - max. # connections to DB2 server
- **MAX_CONNECTIONS** - max. # of concurrent client connections to *DB2 Connect* (default is equal to *MAXAGENTS* - concentrator disabled)
- **MAX_CONNECTIONS > MAXAGENTS** to enable sysplex workload balancing
 - DB2 Connect 8.5 introduced 2 new options
 - `DB2_MAX_CLIENT_CONNRETRIES`
 - `DB2_CONNRETRIES_INTERVAL`



Automatic client reroute configuration: (`DB2_MAX_CLIENT_CONNRETRIES` and `DB2_CONNRETRIES_INTERVAL`)

By default, the automatic client reroute feature retries the connection to a database repeatedly for up to 10 minutes. It is, however, possible to configure the exact retry behavior using one or both of the following two registry variables:

`DB2_MAX_CLIENT_CONNRETRIES`: The maximum number of connection retries attempted by automatic client reroute.

`DB2_CONNRETRIES_INTERVAL`: The sleep time between consecutive connection retries, in number of seconds.

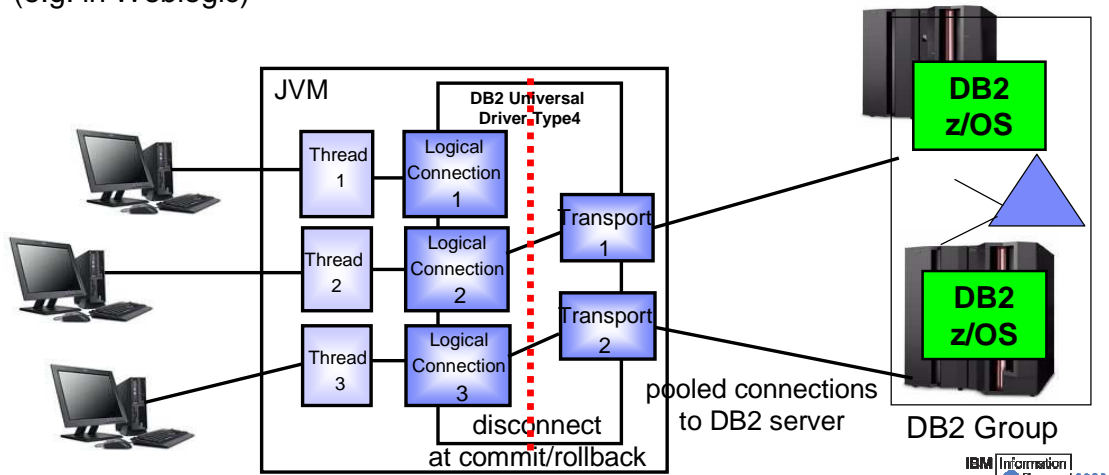
If `DB2_MAX_CLIENT_CONNRETRIES` is set, but `DB2_CONNRETRIES_INTERVAL` is not, `DB2_CONNRETRIES_INTERVAL` defaults to 30.

If `DB2_MAX_CLIENT_CONNRETRIES` is not set, but `DB2_CONNRETRIES_INTERVAL` is set, `DB2_MAX_CLIENT_CONNRETRIES` defaults to 10.

If neither `DB2_MAX_CLIENT_CONNRETRIES` nor `DB2_CONNRETRIES_INTERVAL` is set, the automatic client reroute feature reverts to its default behavior described previously.

JCC Type 4 Sysplex Workload Balancing

- JCC type 4 supports Sysplex Workload Balancing
 - ✓ JDBC 2.0 datasource since DB2 Connect V8 FP10 –>JCC 2.7.xx (e.g. in WebSphere Application Server)
 - ✓ JDBC 1.2 DriverManager since DB2 Connect 9.5 –>JCC 3.50.xx (e.g. in Weblogic)



JCC Type 4 Sysplex Workload Balancing . . .

➤ Typical [DataSource Properties](#)

- **enableSysplexWLB=YES** - enables Sysplex Workload Balancing
 - default is false, disabled
- **enableConnectionConcentrator** → defaults to true when enableSysplexWLB is true
- **maxTransportObjects** - max # of connections to DB2 server from this DataSource.
 - Can not be bigger than `db2.jcc.maxTransportObjects`
 - default value is -1, meaning no limit

IBM Data Server Driver for JDBC and SQLJ data source properties

- enableSysplexWLB
- enableConnectionConcentrator → enableConnectionConcentrator defaults to true when enableSysplexWLB is true
- maxTransportObjects

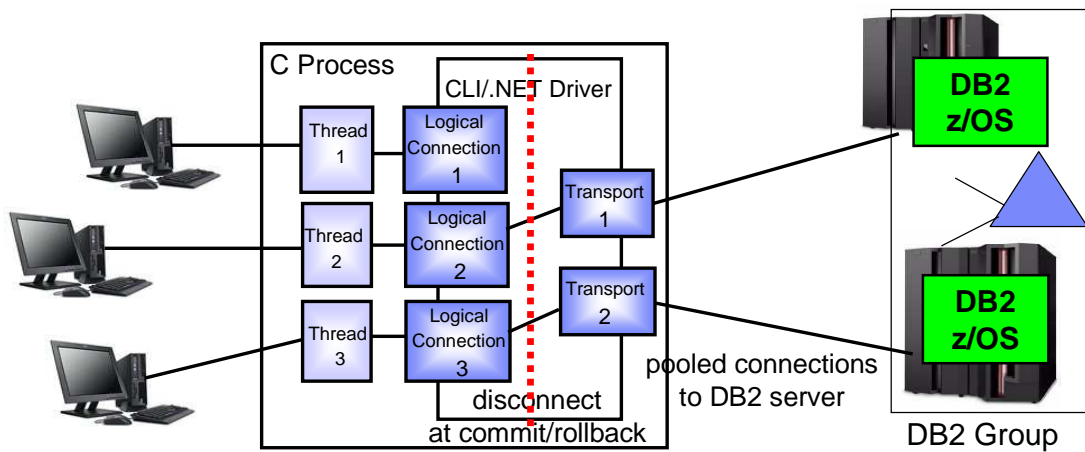
JCC Type 4 Sysplex Workload Balancing . . .

➤ **Global properties** defined in **Global Properties File**:

- **db2.jcc.maxTransportObjects** - max # of connections to DB2 server across all datasources (default value is -1, meaning no limit)
- **db2.jcc.maxTransportObjectIdleTime** - time in sec., a connection stays idle in the pool before it is closed,
 - ☞ default value is 60 sec
- **db2.jcc.maxTransportObjectWaitTime** - if maxTransportObjects is reached - time in sec., an application waits to get a connection before throwing a SQLException
 - ☞ default value is 5 sec
- **blockingReadConnectionTimeout** –
 - ☞ socket timeout value
 - ☞ should be set to a value greater by few seconds than the time it takes to execute the longest query in the application
- **loginTimeout** –
 - ☞ only used for new connection

C-based Client Sysplex Workload Balancing

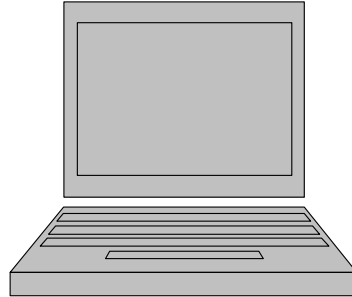
- Data Server Driver 9.5 FP3 for CLI and .NET supports Sysplex Workload Balancing
- Limited XA Support



XA coordination needs to happen on the same connection as the SQL.
 Some XA coordinator handle XA coordination on a different connection than the SQL was flown. You would need to continue to use DB2 Connect Server for those.

XML Config. File – db2dsdriver.cfg

```
< configuration >  
  < DSN_Collection >  
    < dsn alias=, name=, host=, port= >  
    < parameter name=, value= />  
  </dsn >  
</DSN_Collection >  
< databases >  
  < database name=, host=, port=,>  
    < parameter name=, value= />  
    < WLB >  
      < parameter name=, value= />  
    </WLB >  
    < ACR >  
      < parameter name=, value= />  
    </ACR >  
  </database >  
</databases >  
< parameters >  
  < parameter name=, value= />  
</parameters >  
</configuration >
```



WLB Configuration

```
<databases>
  <database name="STLEC1" host="9.30.30.5" port="446">
    <WLB>
      <parameter name="enableWLB" value="true"/>
      <parameter name="maxTransports" value="100"/>
      <parameter name="maxTransportIdleTime" value="600"/>
      <parameter name="maxTransportWaitTime" value="30"/>
      <parameter name="maxRefreshInterval" value="30"/>
    </WLB>
    <ACR>
      <parameter name="enableACR" value="true"/>
      <parameter name="enableSeamlessACR" value="true"/>
    </ACR>
  </database>
</databases>
```

enableAcr or enableSeamlessAcr when using CLI driver, when enableWLB is true that is default

Fault Tolerant Sysplex Support

- **Initial connection to DB2 data sharing group url**
 - recommended to be a distributed DVIPA. This virtual IP address resolves to any available member based on WLM
 - or a DNS name that resolves into a list of IP address of each member that is used round robin
 - It always works if at least 1 member is up.

- **The initial connection (or any ACCRDB responds) returns a member IP list with WLM weights**

- **Any following connection: uses client sysplex workload balancing algorithm to determine which member to use if reuse is OK:**
 - No open WITH HOLD cursor
 - No declared global temporary tables must exist
 - (used declared global temp tables must be explicitly or implicitly dropped)
 - No reference to packages bound with KEEP DYNAMIC YES
 - Rollback always leaves a connection in a reusable state

Behavior at Connection Errors

- **DB2 returns indicator as part of COMMIT responds if transport can be reused and returns SET statements to replay connection state at reuse**
- *If first SQL stmt in transaction fails and reuse OK*
 - No errors reported back to application
 - SET statements associated with the logical connection are replayed with first SQL on another transport
- *If subsequent SQL fails and reuse OK,*
 - -30108 reuse error returned to application (transaction is rolled back and reconnected).
 - SET statements are replayed to recover connection state
 - Up to application to retry transaction
- *If subsequent SQL and reuse not OK*
 - -30081 connection failed error returned to application.
 - Connection returned to initial (default) state
 - application needs to reestablish connection state and retry transaction
- **If all members in the member list are tried and none seems to be available, the initial data source url (DVIPA/DNS name) is retried to make sure that really no member is available.**

WebSphere Connection Pooling

- Connection object pool currently maintained by WebSphere
 - saves creating/destroying Connection objects which is relatively expensive
 - part of the Connection object creation is creating a physical connection to DB2
- Connection object holds references to other Java objects like **preparedStatement** objects that would be destroyed with the Connection object

In WebSphere, Connection Pooling is a function of WebSphere, not a function of the JDBC driver. The Universal JDBC driver does not implement connection pooling. Note that if a DB2 subsystem shuts down, WebSphere connection pools are not notified. If the purge policy is to only purge a thread, multiple transactions will encounter errors until the entire connection pool is purged. On the other hand, an intermittent failure, such as an idle thread will timeout entire pool.

WebSphere Connection Pooling

Data sources > eStoreDataSource > Connection pools

Use this page to set properties that impact the timing of connection management tasks, which can affect the performance of your application. Consider the default values carefully; your application requirements might warrant changing these values.

Configuration

General Properties	Additional Properties
Scope cells:ws1cell:nodes:ws1node9:servers:ws1sr19	Advanced connection pool properties
Connection timeout 180 seconds	Connection pool custom properties
Maximum connections 10 connections	
Minimum connections 4 connections	
Reap time 180 seconds	
Unused timeout 1800 seconds	
Aged timeout 0 seconds	
Purge policy EntirePool	

Apply OK Reset Cancel

WebSphere Connection Pool Properties

Max Connections → max connections from JVM instance

Min Connections → lazy minimum number of connections in pool

Reap Time → How often cleanup of pool is scheduled in seconds

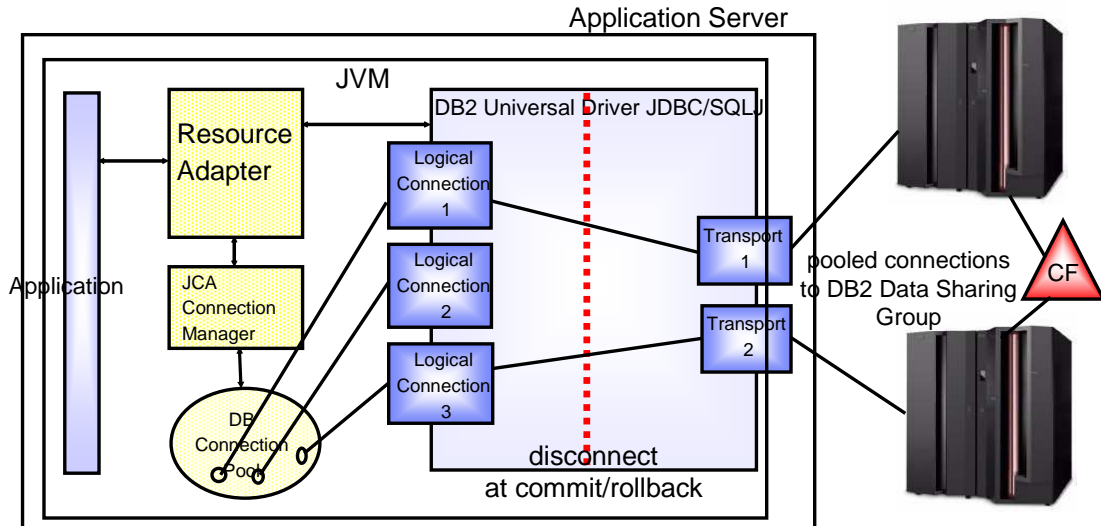
Unused Timeout → How long to let a connection sit in the pool unused

Aged Timeout → How long to let a connection live before recycling

Purge Policy → After StaleConnection, does the entire pool get purged or only individual connection

WebSphere Application Server - End-to-end Example

- Always use connection pool closest to application
- Enable Sysplex Workload Balancing for best availability



WAS/DB2 Active Thread - Tuning Considerations

- **WAS connections in connection pool that keep an active thread in DB2 are target of the "idle thread timeout"**
 - Type 2 on z/OS (RRSAF)
 - Type 4 connection that holds on to resources, e.g. WITH HOLD cursor, KEEP DYNAMIC YES.

- **Set WAS "connection unused time" to a smaller value than DB2 "idle thread timeout" to avoid stale connection conditions.**

- **Consider setting "min connections" to 0 (zero) and "connection unused time" to not higher than 10 - 15 min to free up unused resources in DB2 in a controlled way and to reduce the exposure of long living threads**

Important DB2 zparm Values

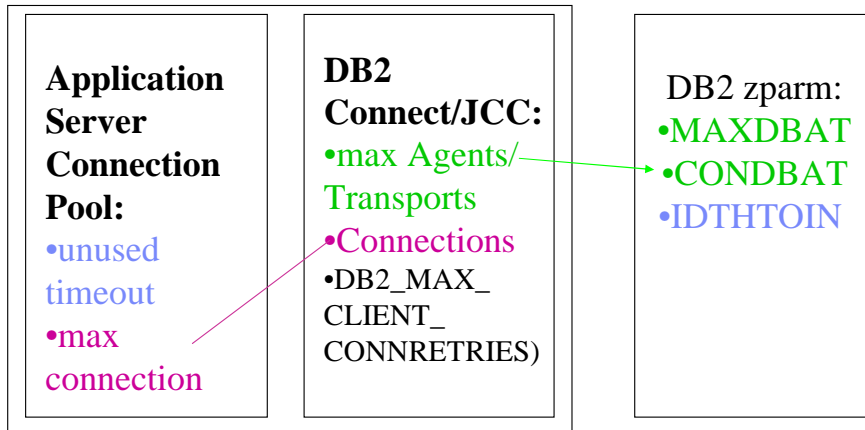
- **CONDBAT** – Max. # of distributed connection into DB2 system
 - includes inactive and active connections, may be large
 - DB2 queues DBAT requests to become active up to CONDBAT

- **MAXDBAT** - Max # database access threads (DBATs) that can be active concurrently.
 - In many installations, max. value determined by available storage in DBM1 (check IFCID 225)
 - Set this value conservatively

- **CMTSTAT INACTIVE** – make a thread inactive after it successfully commits or rolls back and thread does not hold resources
 - prerequisite for sysplex workload balancing
 - inactive connections use less storage and free up DBM1 resources

- **IDTHTOIN** – time in sec an active server thread remain idle before it is canceled
 - inactive connections are not subject to idle thread timeout
 - Strongly recommended to not set to 0 – disable, default works well

WAS/DB2 Inactive Thread - Tuning Considerations



Important Parameters:

DB2 z/OS

zparm MAXDBAT, CONDBAT, Idle thread timeout

DB2 Connect

Set the 6th positional parameter of the DCS directory entry representing the DB2 group to: ,,,,SYSPLEX

MAXAGENTS, NUM_POOLAGENTS, Connections,

Since V8.2 DB2_MAX_CLIENT_CONNRETRIES

JCC Type 4

db2.jcc.maxTransportObjects, db2.jcc.maxTransportObjectIdleTime

, db2.jcc.maxTransportObjectWaitTime

Application Server

#max connections, unused timeout

Closing recommendations ...

➤ Application Enablement

- Enable inactive thread support for the DB2 group
- Minimize the use of KeepDynamic
 - Static SQL most efficient use of resources
- Close result sets and held cursors when done
- Drop global resources if not needed anymore
- Timely commits
- Allows for more efficient use of group resources

In order to fully utilize a DB2 group, it is important that applications do not hold resources or threads for long periods of time.

The DB2 Connect Server makes a connection available to other applications as soon as the application commits a transaction and does not hold any global resources that are required to persist across transactions.

For example:

- any WITH HOLD cursors not closed
- any declared global temp tables not dropped
- any application using packages bound using the KEEP DYNAMIC option

Any of these resources held across a commit would prevent the connection and associated DB2 thread from being pooled. Preventing it to be reused by other applications reducing the overall efficiency of the DB2 subsystem. If the transaction completes and no global resources are open, the DB2 Connect Server can disassociate the transport from one application logical connection and associate the transport for another application logical connection.

Closing recommendations ...

- **Best practices, high available, environment for strategic, enterprise applications involves:**
 - DB2 data sharing group enabled with DVIPA and sysplex distributor
 - Optionally, subsetting let's you focus applications on a subset of servers
 - Application servers/DB2 Client drivers enabled for sysplex workload balancing
 - Applications are well-behaved

Reference

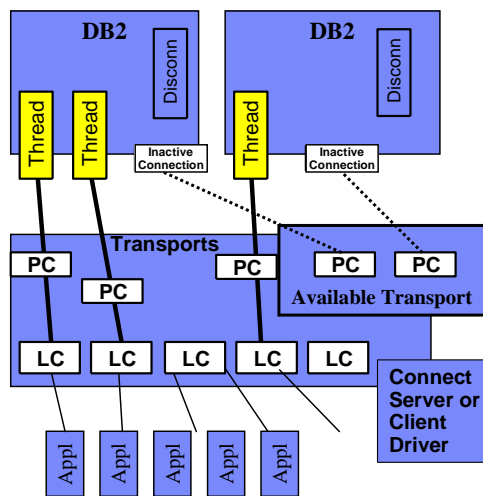
- **DB2 9 for z/OS Data Sharing: Distributed Load Balancing and Fault Tolerant Configuration**
<http://www.redbooks.ibm.com/abstracts/redp4449.html?Open>

- **DB2 9 for z/OS: Distributed Functions (SG24-6952)**
<http://www.redbooks.ibm.com/abstracts/sg246952.html>

- **DB2 Version 9.1 for z/OS Data Sharing: Planning and Administration (SC18-9845)**
http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp?topic=/com.ibm.db29.doc.dshare/db2z_dshare.htm

Backup slides

DB2 transaction level load distribution



- DB2 client driver maintains pool of physical connections (PC) to members called transports to DB2 subsystems based on WLM recommendations
- Applications establish logical connections (LC) to DB2 client driver
- Connections not associated with a DB2 thread called inactive connections by DB2
- Threads not currently processing work are DISCONNECTED DBATs
- Connections are associated with a subsystem transport on transaction boundaries based on WLM recommendations

As I said before, this function dramatically increases scalability of the DB2 group across the parallel sysplex. Let me go into some detail on how transaction load distribution works:

DB2 Connect Server maintains a pool of physical connections to each subsystem, shown in the diagram as (PC). These are called transports. The number of transports to each subsystem is based on WLM recommendations. When a transport is not being used and is in the available transport pool then it shows up on the DB2 server as a disconnected connection. There is no DB2 thread is associated with the connection.

Each DB2 subsystem can theoretically support 125,000 disconnected connections

Applications establish logical connections, shown in the diagram as (LC). The DB2 Connect Server assigns a transport only when a transaction is initiated on the This allows applications to stay connected without any resources being consumed on the DB2 server. It is now possible to have thousands of users to be active in an application(s) and yet only have a few threads active on DB2.

Another key feature is the method of assigning a transport. The DB2 Connect implements a sophisticated scheduling algorithm that uses WLM recommendations to distribute workload across the subsystems. WLM is not only aware of the load on each member but also their availability. This allows the DB2 Connect Server to transparently relocate work away from failed or overloaded members to those that are up and under utilized.

Transports/Agents – Threads to DB2 . . .

- **Light reuse**
 - connection checks for the availability of its last used transport first (DB2 Connect Server only does this up to 8 times)
- **Medium reuse**
 - if not available, it checks for the last recently used transport (FIFO) to give other connections a chance to find their last used transport to optimize for light reuse
 - If the transport was previously used by the same application, only SET properties flow with first SQL
- **Heavy reuse**
 - if the connection is reused by a different application (set of attributes) reset, SET properties and first SQL chained flow on the first message
- **Transports are closed when they are idle for a “maxTransportObjectIdleTime” time in transport pool (DB2 Connect Server does not have a configurable idle time)**
- **Connection receives an exception after “maxTransportObjectWaitTime” time waiting for a transport to become available (not sure of DB2 Connect Server)**
- **Refresh member list**
 - if a member list was not updated for 30 seconds, the next transaction flows a heavy reuse flow